

OBJECT

Tim Ajar Matakuliah Dasar Pemrograman 2022

Konsep Pemrograman Berorientasi Objek

- Java → merupakan bahasa pemrograman yang secara penuh menerapkan konsep PBO
- PBO → paradigma pemrograman yang memandang suatu **program** terdiri dari **kumpulan objek-objek** yang saling **berinteraksi**
- **Objek** menjadi perhatian utama
- Ketika hendak membuat suatu program dari suatu, yang harus diidentifikasi dulu adalah **objek-objek** apa yang ada di dalam sistem tersebut

Konsep Pemrograman Berorientasi Objek

	PBO	Prosedural
Sudut pandang	Melihat program sebagai kumpulan objek2 yg berinteraksi	Melihat program sebagai kumpulan proses2/prosedur2
Fokus utama	Objek	Proses/Prosedur/Fungsi
Contoh 1	Sistem Perkuliahan di Kelas: <ul style="list-style-type: none">• Mahasiswa• Dosen• Matakuliah• Form Absensi• Surat ijin• Soal	Sistem Perkuliahan di Kelas: <ul style="list-style-type: none">• Bertanya• Menjawab• Presentasi• Mengabsensi kehadiran• Mengerjakan soal

Konsep Pemrograman Berorientasi Objek

	PBO	Prosedural
Contoh 2	Sistem Perbankan: <ul style="list-style-type: none">• Akun/Rekening• Nasabah• Uang	Sistem Perbankan: <ul style="list-style-type: none">• Setor• Tarik• Transfer

Konsep Pemrograman Berorientasi Objek

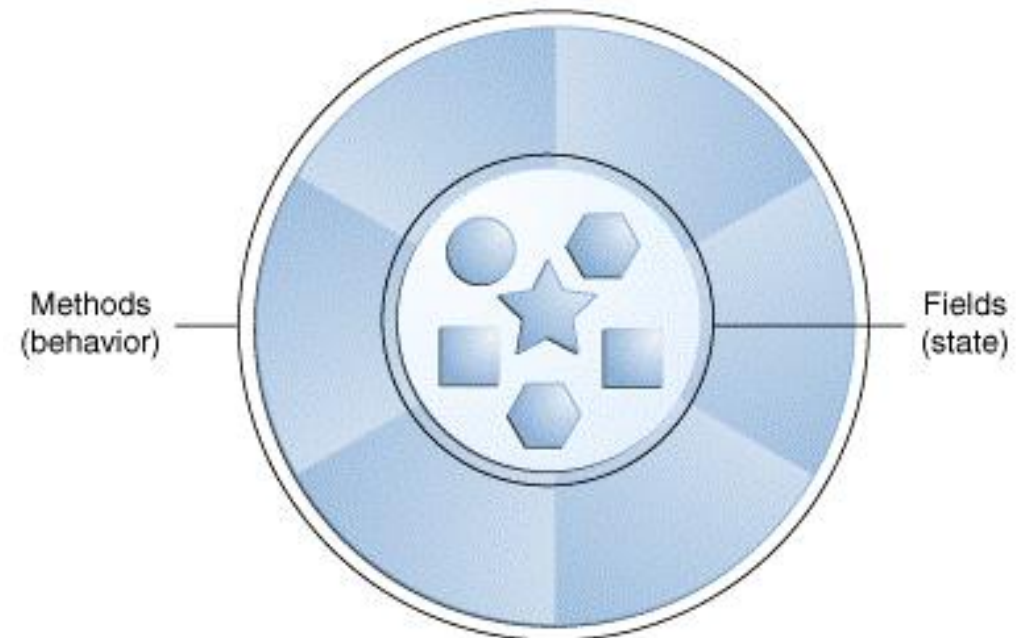
- Konsep PBO yang paling mendasar adalah (1) Class dan (2) Object
- Selain kedua konsep dasar tersebut, ada beberapa konsep dasar lain:
 - Enkapsulasi
 - Inheritance
 - Polimorfisme
- Pada semester ini, konsep yang akan dominan digunakan adalah Class dan Object
- Konsep Enkapsulasi, Inheritance, dan Polimorfisme akan dijelaskan detil di matakuliah PBO

Object

- Pada slide sebelumnya disebutkan bahwa fokus utama **PBO** adalah pada **Object**
- **Object** merupakan benda **nyata**
- Contoh objek yang ada di dalam kelas:
 - Meja 1, Meja 2 ... Meja 30
 - Mahasiswa 1, Mahasiswa 2...Mahasiswa 30
 - LCD
 - Papan tulis
 - Laptop, dst

Object

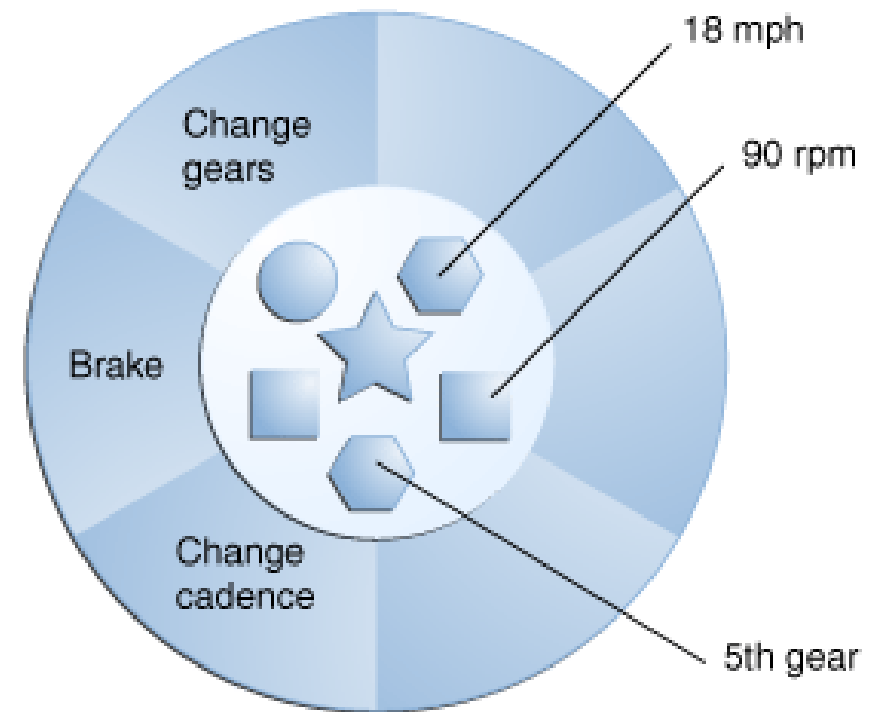
- Karakteristik **Object**:
 - Memiliki sesuatu data/karakter/properti/variabel/state/field/**atribut**
 - Bisa melakukan sesuatu tingkah laku/behaviour/fungsi/**method**



Object

- Contoh **Object**:

Object <u>Sepeda</u>	
Atribut	Method
- Gear = 5	- Ubah gear
- Kecepatan = 18	- Mengerem
- Rpm = 90	- Ubah ritme





Object

- Contoh **Object**:

Object <u>Mahasiswa</u>	
Atribut	Method
- NIM = 1614210045	- Mengikuti ujian
- Nama = John Doe	- Melihat KHS
- IPK = 3.5	- Melihat tunggakan akademik
- Alamat = Malang	



Class

- Setiap **object** pasti berawal/dibuat dari suatu rancangan/desain/template/class
- Class merupakan **template** untuk membuat object
- Proses pembuatan object dari suatu class disebut **instansiasi**
- Object tidak akan ada jika tidak ada class, dan class tidak akan bisa digunakan jika belum dibuat objek nya.



Class

	Class	Object
Penjelasan	Masih berupa rancangan/template/desain/blueprint	Objek nyata/riil yang sudah dibentuk dari suatu class
Sifat	Umum	Spesifik
Contoh 1	<ul style="list-style-type: none">• Mahasiswa• Dosen• Matakuliah	<ul style="list-style-type: none">• Mahasiswa 1, Mahasiswa 2, dst• Dosen A, dosen B, dst• Matakuliah PBO, matakuliah basis data

Class

- Apakah **Class** juga **memiliki atribut dan method** ?
- Ya, Class juga memiliki atribut dan method, sebagaimana objek juga memilikinya. Bedanya, jika atribut dan method yang di objek sudah riil, sudah bisa diberi nilai tertentu. Sedangkan atribut dan method di class masih berupa rancangan.



Class

- Format Deklarasi Class:

```
class NamaClass{  
    //deklarasi atribut  
    //deklarasi method  
}
```

- Contoh:

```
class Sepeda{  
  
}
```

Atribut

- Atribut : data yang dimiliki oleh object atau class
- Nama atribut biasanya ditandai dengan **kata benda**
- Contoh : Atribut yang dimiliki oleh mahasiswa antara lain nim, nama, IPK, alamat



Atribut

- Format Deklarasi Atribut :

```
tipeData namaAtribut;
```

- Contoh:

```
String nama;  
float ipk;  
int umur;  
boolean statusLulus;
```



Method

- Method : merupakan proses/tingkah laku/fungsi yang bisa dilakukan oleh suatu objek
- Method digunakan oleh objek untuk berinteraksi dengan objek yang lain
- Nama method biasanya ditandai dengan **kata kerja**
- Contoh : method melihat KHS, mengerjakan ujian dari objek mahasiswa



Method

- **Format Deklarasi Method:**

```
tipeData namaMethod(tipeData param) {  
    //isi method  
}
```

- **Contoh:**

```
void tampilStatus(boolean st) {  
    if(st==true)  
        System.out.println("Lulus");  
    else  
        System.out.println("Belum Lulus");  
}
```



Instansiasi

- Instansiasi → Proses pembuatan objek dari suatu class
- Ditandai dengan kata kunci **new**
- Contoh:

```
Sepeda spd = new Sepeda ();
```



Mengakses Atribut dan Method suatu Objek

- Setelah objek terbentuk, selanjutnya atribut dan method yang dimiliki objek tersebut bisa mulai digunakan atau diakses.
- Cara mengakses atribut:

```
namaObjek.namaAtribut = nilai;
```

- Cara mengakses method:

```
namaObjek.namaMethod();
```

Contoh:

```
spd.kecepatan = 10;  
System.out.println(spd.kecepatan);  
spd.tampil();
```



Ilustrasi Program

```
1 public class Sepeda { Deklarasi Class
2     float kecepatan;
3     int gear; Atribut
4
5     void tambahKecepatan(float a){ Method
6         kecepatan=kecepatan+a;
7         sesuaikanGear();
8     }
9     void kurangiKecepatan(float a){
10         kecepatan=kecepatan-a;
11         sesuaikanGear();
12     }
13     void sesuaikanGear(){
14         if(kecepatan<10){
15             gear=1;
16         }else if(kecepatan>=10 && kecepatan<20){
17             gear=2;
18         }else if(kecepatan>=20 && kecepatan<40){
19             gear=3;
20         }else{
21             gear=4;
22         }
23     }
24     void tampil(){
25         System.out.println("Kecepatan="+kecepatan+", Gear="+gear);
26     }
27 }
```



Ilustrasi Program

```
1 public class SepedaMain {  
2     public static void main(String[] args) {  
3         Sepeda spd = new Sepeda();  
4         spd.kecepatan = 10;  
5         spd.tambahKecepatan(5);  
6         spd.tampil();  
7     }  
8 }
```

Instansiasi

Memberikan nilai
atribut dari objek **spd**

Menjalankan method dari objek **spd**



Konstruktor

- Konstruktor → Method **istimewa** yang digunakan saat **pembuatan objek (instansiasi)**
- Istimewa:
 - Nama method sama dengan nama class
 - Tidak memiliki tipe data method
 - Hanya bisa dijalankan/dipanggil pada proses instansiasi
 - Bisa memiliki parameter



Konstruktor

- Jenis konstruktor
 - Konstruktor Default :
 - konstruktor yang tidak memiliki parameter
 - konstruktor default
 - Konstruktor Berparameter : konstruktor yang memiliki parameter
- Contoh konstruktor default:

```
public Sepeda() {  
}
```

- Contoh konstruktor berparameter

```
public Sepeda(float kecp, int ge) {  
}
```



Konstruktor

- Konstruktor digunakan pada saat instansiasi
- Contoh penggunaan konstruktor default:

```
Sepeda spd = new Sepeda();
```

- Contoh penggunaan konstruktor berparameter:

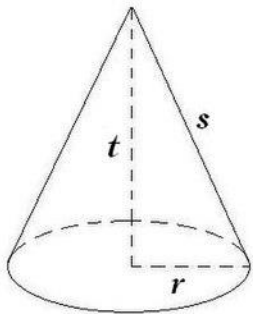
```
Sepeda spd = new Sepeda(15.5, 2);
```


Diagram Class

Sepeda
kecepatan: float gear: int
tambahKecepatan(): void kurangiKecepatan(): void sesuaikanGear(): void tampil(): void

Latihan

1. Tentukan sebuah objek yang ada di sekitar anda, tentukan **tiga** atribut dan **dua** method yang dimiliki oleh objek tersebut!
2. Sebuah bangun ruang berbentuk kerucut memiliki tinggi dan jari-jari. Kerucut tersebut dapat dihitung volume dan luas permukaannya. Buatlah diagram class untuk kerucut!



3. Rancanglah diagram class untuk Anggota Perpustakaan!